A Reconfigurable Multiprocessor Architecture and its Arithmetic Performance

Kenneth B. Winiecki, Jr.

Department of Electrical and Computer Engineering
Clemson University

July 22, 1994

Overview

1. Background Information
2. Prior Development
3. Processing Element Architecture
4. Arithmetic Performance
5. Conclusion

1

Parallelism

•Form
- –multicomputing
- –pipelining
- –multiprocessing √

•Function
- –SISD
- –MISD
- –SIMD
- –MSIMD √
- –MIMD

1

Reconfigurability

The ability to vary the number of processors applied to an operation.

Example:

PEs	Ops	PEs / Op	Virt-PEs	Cyc / Op	Tot Cyc	
4000	2000	1	4000	300	300	
4000	2000	2	2000	200	200	√
4000	3000	1	4000	300	300	√
4000	3000	2	2000	200	400	

Enables performance increase through tradeoff between time per operation and number of simultaneous operations.

1

Reconfigurability

- Form
  - hardware
  - software √
- Function
  - precision
  - method √

2

Progenitor

- DAP, MPP, CM–1
- Process Large Arrays of Homogeneous Data
- Massively Parallel SIMD
- Precision Reconfigurable
- Rectangular Mesh Interconnection

2

Developments

- *A Bit-Serial SIMD Processing Element* (Ligon)
  - improve CM–1
  - 3 clock cycles ⇨ 1 clock cycle
  - memory-to-memory ⇨ register-to-register
  - ⇨ added parallel memory-register transfer

- SIMD Architectures Should Support Floating-Point

- *An Empirical Evaluation of Architectural Reconfigurability* (Ligon)

2

SIMD Evolution Alternative

- Use Word-Wide PEs

- Design New PE from previous 1-Bit PE

- Use Carry-Lookahead Logic and Variable-Shift Registers

- Support Method Reconfigurability

- Use the MSIMD Paradigm

2

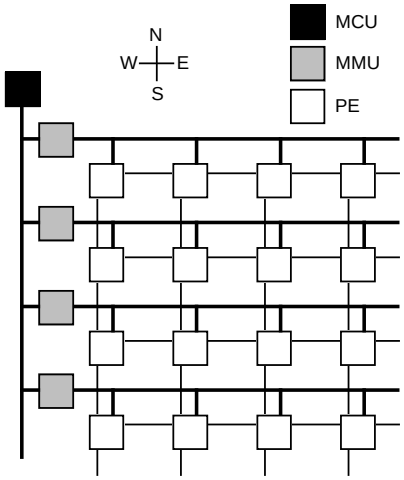
Tasks

Objective: Evaluate These Architectural Characteristics

- Design and Model PE behavior in Verilog HDL
- Design and Model Configurations for Arithmetic in Verilog HDL
  - integer and floating-point operations
  - various numbers of PEs
  - various data word widths
- Analyze Results

2

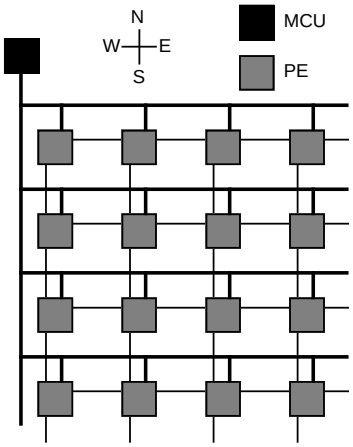
System Architecture



Global Data Routing System (not shown)
with connections to each PE

2

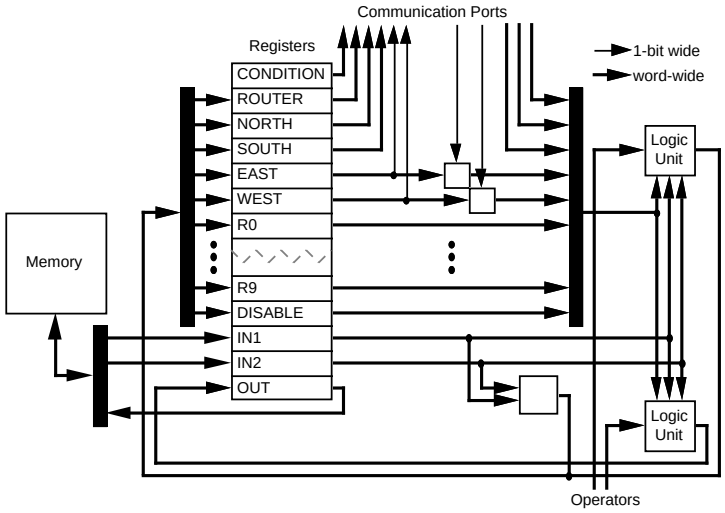
Alternate System Architecture



Global Data Routing System (not shown)
looks exactly like MCU

3

Processing Element



Control Unit and Microinstruction Memory not shown

3

Boolean Logic Units

•Boolean Logic Units are banks of W 1-of-8 multiplexers

•Inputs are 8-bit coded operators from microinstruction

•Controls are W-bit operands

- IN1
- IN2
- selectable source register

3

BLU Example

Bit Number	IN1 Bit	IN2 Bit	Source Bit	Function: IN2	Function: IN1 ^ SRC
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	1
5	1	0	1	0	0
6	1	1	0	1	1
7	1	1	1	1	0
				Selectable Destination Operator	OUT Destination Operator

3

Verilog–XL Behavioral HDL

- structured procedures for sequential or concurrent execution
- explicit control of the time of procedure activation specified by both delay expressions and by value changes called event expressions
- explicitly named events to trigger the enabling and disabling of actions in other procedures
- procedural constructs for conditional, if-else, case, and looping operations
- procedures called tasks that can have parameters and non-zero time duration
- procedures called functions that allow the definition of new operators
- arithmetic, logical, bit-wise, and reduction operators for expressions

4

Speedup

•Arithmetic operations are fixed-size problems

•Absolute Speedup: $S[P] = T[1] \div T[P]$

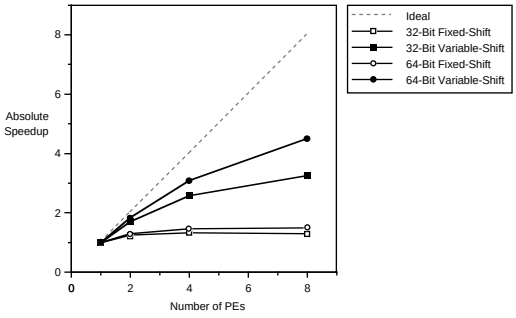
•Relative Speedup: $S'[P] = S[P] \div P$

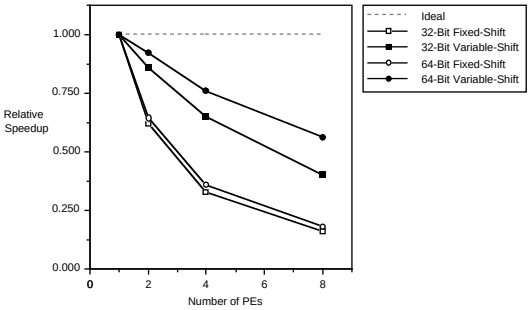
4

Integer Multiplication

Word Width	Number of PEs	Variable Shift	Loop Cycles	Other Cycles	Total Cycles	Absolute Speedup	Relative Speedup
32	1	N/A	4	14	142	1.00	1.00
32	2	No	6	19	115	1.23	.62
32	4 *	No	10	27	107	1.33	.33
32	8 *	No	18	40	112	1.26	.16
32	2	Yes	4	19	83	1.71	.86
32	4	Yes	4	23	55	2.58	.65
32	8	Yes	4	28	44	3.23	.40
64	1	N/A	4	14	270	1.00	1.00
64	2	No	6	19	211	1.28	.64
64	4 *	No	10	27	187	1.44	.36
64	8 *	No	18	40	184	1.47	.18
64	2	Yes	4	19	147	1.84	.92
64	4	Yes	4	23	87	3.10	.76
64	8	Yes	4	28	60	4.50	.56

* calculated

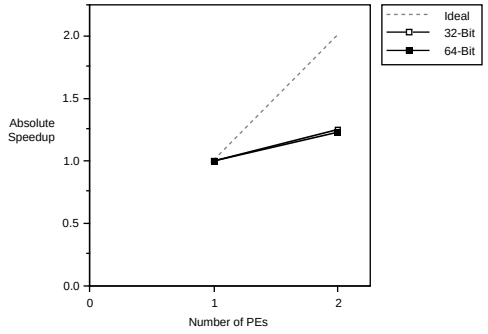


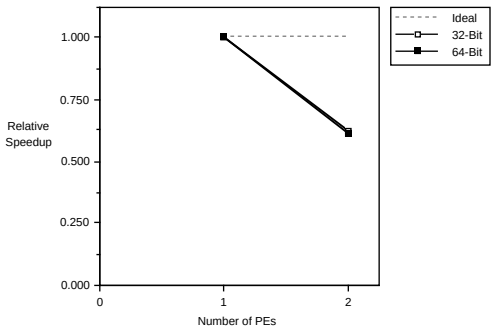


4

Integer Division

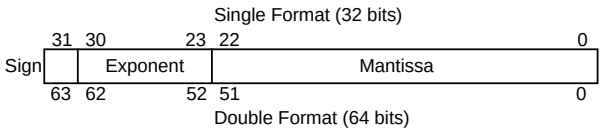
Word Width	Type	Number of PEs	Loop Cycles	Other Cycles	Total Cycles	Absolute Speedup	Relative Speedup
32	R	1	6	21	213	1.00	1.00
32	R	2	5	12	172	1.24	.62
32	NR	1	6	25	217		
32	NR	2	5	15	175		
64	R	1	6	21	405	1.00	1.00
64	R	2	5	12	332	1.22	.61
64	NR	1	6	25	409		
64	NR	2	5	15	335		





4

Floating-Point Format



- IEEE Standard
  - mantissa sign bit
  - $2^{n-1}-1$ biased 2s-complement signed exponent
  - normalized mantissa magnitude fraction
- Deviations
  - no additional bits
  - no extended formats
  - no NaN, ∞, denormalized values, negative zero
  - truncation is only rounding method used

4

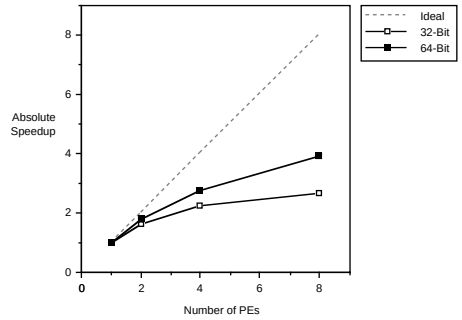
Floating-Point Addition/Subtraction

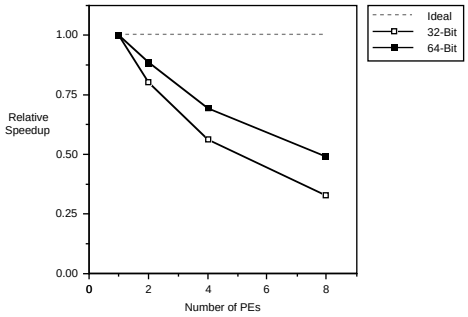
Word Width	Number of PEs	Alignment Loop Cycles—Iterations		Normalization Loop Cycles—Iterations		Other Cycles	Total Cycles
32	1	4	24	3	25	33	204
64	1	4	53	3	54	33	407

4

Floating-Point Multiplication

Word Width	Number of PEs	Loop Cycles	Loop Iterations	Other Cycles	Total Cycles	Absolute Speedup	Relative Speedup
32	1	5	24	24	144	1.00	1.00
32	2	5	12	30	90	1.60	.80
32	4	5	6	34	64	2.25	.56
32	8	5	3	39	54	2.66	.33
64	1	5	53	24	289	1.00	1.00
64	2	5	26	35	165	1.78	.88
64	4	5	13	39	104	2.78	.69
64	8	5	6	44	74	3.91	.49





5

Conclusion

- Designed and Modeled PE and Arithmetic Configurations
- Showed No Problems Performing Arithmetic with This PE
- Showed Reconfigurability Does Facilitate Parallelization and Speedup for Many Operations
- Indicated that MSIMD Has Potential to Yield a High Performance Machine

5

Future Directions

- Experiments should be repeated without the carry/logic exclusion rule
- Floating-point division should be investigated
- Floating-point addition should be completed
- More research on integer division may be beneficial
- Other operations should be investigated (logarithms, etc.)
- Experimenting with configurations for original MSIMD architecture may be beneficial
- Microcontroller and microinstruction memory should be merged PE model
- An addressing mode other than direct should be added to the PE model
- Next significant step is to move from behavioral models to structural models and simulate a whole system